ECSE-6610: PR Homework Set 2

Chengjiang Long

February 23, 2018

Assigned Date: Feb 13, 2018.
Due Date: Feb 25, 2018.
Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited.
Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.
Submission Format. Electronic submission of a zip file is mandatory. Include code in your pdf file as needed to make your answers clear. Submit all code separately.

Problem 1 (30 points) Explore some of the properties of density estimation in the following way.

(a) [**5 points**] Write a program to generate points according to a uniform distribution in a unit cube, $-1/2 \leq x_i \leq 1/2$ for $i = 1, 2, 3$. Generate 10^4 such points.

(b) [**5 points**] Write a program to estimate the density at the origin based on your 10^4 points as a function of the size of a cubical window function of size h. Plot your estimate as a function of h, for $0 < h \leq 1$.

(c) [**5 points**] Evaluate the density at the origin using n of your points and the volume of a cube window which just encloses k points. Plot your estimate as a function of $k = 1, ..., 10^4$.

(d) [**10 points**] Write a program to generate 10^4 points from a spherical Gaussian density (with $\mathbf{\Sigma} = \mathbf{I}$) centered on the origin. Repeat (b) and (c) with your Gaussian data.

(e) [**5 points**] Discuss any qualitative differences between the functional dependencies of your estimation results for the uniform and Gaussian densities.

Problem 2 (15 points) The following three-dimensional data in Table 1 covers three categories, denoted ω_i. Consider k-nearest-neighbor density estimations in different numbers of dimensions:

(a) [**5 points**] Write a program to find the k-nearest-neighbor density for n (unordered) points in one dimension. Use your program to plot such a density estimate for the x_1 values in category ω_3 in the table above for $k = 1, 3$ and 5.

(b) [**5 points**] Write a program to find the k-nearest-neighbor density estimate for n points in two dimensions. Use your program to plot such a density estimate for the (x_1, x_2) values in ω_2 for $k = 1, 3$ and 5.

Table 1: The dataset used in Problem 2

	ω_1			ω_2			ω_3		
sample	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
1	0.28	1.31	-6.2	0.011	1.03	-0.21	1.36	2.17	0.14
2	0.07	0.58	-0.78	1.27	1.28	0.08	1.41	1.45	-0.38
3	1.54	2.01	-1.63	0.13	3.12	0.16	1.22	0.99	0.69
4	-0.44	1.18	-4.32	-0.21	1.23	-0.11	2.46	2.19	1.31
5	-0.81	0.21	5.73	-2.18	1.39	-0.19	0.68	0.79	0.87
6	1.52	3.16	2.77	0.34	1.96	-0.16	2.51	3.22	1.35
7	2.20	2.42	-0.19	-1.38	0.94	0.45	0.60	2.44	0.92
8	0.91	1.94	6.21	-0.12	0.82	0.17	0.64	0.13	0.97
9	0.65	1.93	4.38	-1.44	2.31	0.14	0.85	0.58	0.99
10	-0.26	0.82	-0.96	0.26	1.94	0.08	0.66	0.51	0.88

(c) **[5 points]** Write a program to form a k-nearest-neighbor classifier for the three-dimensional data from the three categories in the table above. Use your program with $k = 1, 3$ and 5 to estimate the posterior probabilities and predict a label for the following points: $(-0.41, 0.82, 0.88)^t$, $(0.14, 0.72, 4.1)^t$ and $(-0.81, 0.61, -0.38)^t$.

Problem 3 (10 points) Describes Fisher's linear discriminant. How is it used to discriminate between data from two classes.

Suppose each datapoint $\mathbf{x}$ in the first class is of form $\mathbf{x} = (x_1, ..., x_{2M})$ where the x_i are i.i.d. from a Gaussian with zero mean and standard deviation σ. The datapoints in the second class are of form $\mathbf{x} = (x_1, ..., x_M, \rho + x_{M+1}, ..., \rho + x_{2M})$, where ρ is fixed and the x_i are also generated by a Gaussian with zero mean and standard deviation σ.

What is Fisher's linear discriminant between these two datasets? Does the discriminant change if ρ is a random variable with distribution $P(\rho)$?

Problem 4 (45 points + Extra 20 points) As we saw in the case of faces, principal component analysis provides a way of creating an optimal low-dimensional representation of a dataset. Now, let's do such a PCA analysis on handwritten digits. Download the dataset from my Google Drive:
train: https://drive.google.com/open?id=1QHpu5xfbKxHIWYVH7BqCFWNArQ5Fgxs7
test: https://drive.google.com/open?id=18Y4aLI2VIZ2eH6FQhSJqyej-8viTeCVZ

Note that this is a subset of the LeCun's MNIST dataset containing just the digits 0, 1, and 2. The full dataset is available at http://yann.lecun.com/exdb/mnist. The dataset is split into training and testing pictures. Do all PCA analysis on the training pictures, and reserve the testing pictures until the last part (nearest neighbor classification). For convenience, I named each image as "img_[*image id number*]_lb_[*image label*].png".

(a) **[10 points]** Write a function to perform PCA on a group of images. This will require you to vectorize the images (i.e., do not do IMPCA). Input the number of dimensions k you want to estimate and output the set of eigenvectors and their corresponding eigenvalues (for the largest k).

(b) [**5 points**] Use the PCA function from question 1 to compute the Digit-0-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Describe what you find in both plots.

(c) [**5 points**] Use the PCA function from question 1 to compute the Digit-1-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Describe what you find in both plots.

(d) [**5 points**] Use the PCA function from question 1 to compute the Digit-2-Space. Plot the mean image and then the first 20 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension (for the first 100 dimensions). Compare and constrast what you find in these plots to the ones you created in questions 2 and 3.

(e) [**10 points**] Implement a nearest-neighbor (NN) classifier to input a training dataset, compute the PCA space, and then take a query image and assign it the class of its nearest neighbor in the PCA space.

(f) [**10 points**] Use the NN classifier to classify the testing images. Prepare a figure that shows 5 correctly classified images of each class and 5 incorrectly classified images of each class. Prepare a table giving the quantitative results over all of the testing data. Explain your findings.

(g) [**Extra 20 points for open solutions**] Implement another two classifiers and then compare their performances with the NN classifier. Also, please try to use at least two distance measurements to see the impact of different distance metrics.